SOFTMAIL®

การเขียนโปรแกรมจัดการข้อมูลแบบ BROWSE สำหรับแก้ไข และป้อนข้อมูล

ยืน ภู่วรวรรณ

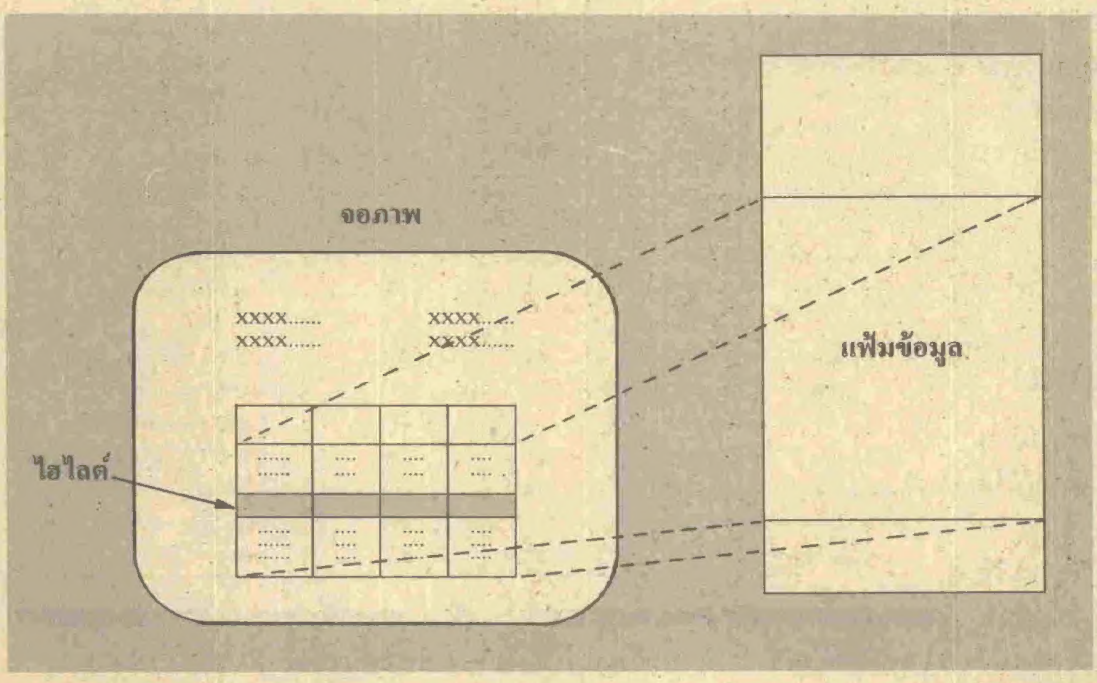
ท่านลองนึกถึงแฟ้มข้อมูลหนึ่งแฟ้ม เมื่อท่านต้องการเขียนโปรแกรมแก้ไขข้อมูลโดยทั่วไป ก็จะใช้วิธีเรียกมาทีละเรคอร์ดบนจอภาพแล้วแก้ไข การแก้ไขในรูปแบบที่ละเรคอร์ดดูจะยังไม่สะดวกนัก น่าจะมีวิธีการนำข้อมูลมาแสดงบนจอภาพเหมือนตาราง ๆ หนึ่งแล้วท่านเลื่อนไปแก้ไขที่ใดก็ได้ตามต้องการ วิธีการนี้ก็คือ การ BROWSE นั่นเอง ดีเบสใช้คำสั่ง browse ได้แต่จะใช้ขณะอยู่ใน . และไม่สามารถเลือกฟิลด์ หรือช่องสกรีนตามความต้องการ ดังนั้น เราจึงต้องหาวิธีการเขียนโปรแกรมให้มีการทำงานเหมือน browse แต่กำหนดสิ่งแวดล้อมได้ตามต้องการ

ความสำคัญของบทความนี้อีกจุดหนึ่งคือ การป้อนข้อมูล แก้ไขข้อมูลที่เป็นบรรทัด เช่น เราต้องการป้อนข้อมูลใบส่งของ ใบเสร็จ หรือเอกสารทางธุรกิจที่มีลักษณะเป็นตาราง โปรแกรมนี้จะช่วยทำให้การป้อนข้อมูลได้ตามความรู้สึกของผู้ใช้ ทำให้เข้าใจหรือใช้งานได้ง่าย

โครงสร้างและหลักการของ BROWSE

ลักษณะของการนำข้อมูลมาแสดงผลบนจอจะมีลักษณะดังรูปที่ 1 ซึ่งจะนำข้อมูลบางส่วนมาแสดงบนจอภาพ

ในการใช้โปรแกรมช่วยจัดการเราทำให้บางส่วนของจอภาพเสมือนเป็นช่องหน้าต่างที่ส่องไปยังแฟ้มข้อมูล และจะปรากฏเป็นไฮไลต์สำหรับเลือกทีละ



รูปที่ 1 โครงสร้างของ BROWSE

บรรทัด เพื่อแก้ไขข้อมูลเหมือนในคำสั่ง browse ที่ดีเบสได้จัดการให้แล้ว

คำสั่ง SAY....GET ยังไม่เพียงพอกับการใช้ใน BROWSE

หากจะเขียนโปรแกรมทำงาน browse เพื่อดึงข้อมูลมาแก้ไขบนจอภาพ เราจะต้องใช้คำสั่งการอ่านและแสดงผลแบบกำหนดสกรีนได้ ดีเบสมีคำสั่ง SAY ....GET ซึ่งจะอ่านข้อมูลในแฟ้มข้อมูลมาแสดงผลเป็นตารางด้วยคำสั่ง say ได้ในทำนองเดียวกันก็รับการแก้ไขข้อมูลเรคอร์ดใดเรคอร์ดหนึ่งที่ตำแหน่งใดก็ใช้คำสั่ง get ดังนั้น ถ้าจะให้ไฮไลต์กำเนิดขึ้นที่ใดก็ get ณ ตำแหน่งนั้น เมื่อ get เสร็จก็เลื่อนตำแหน่งไป รูปแบบของการทำงานเป็นดังรูปที่ 2

ด้วยหลักการตามรูปที่ 2 เราจึงเลือกเรคอร์ดที่ต้องการมาแสดงด้วย say และให้ไฮไลต์ตำแหน่งใดด้วย get ไฮไลต์เลื่อนไปเมื่อป้อนเสร็จเรคอร์ดหนึ่ง ๆ แต่ปัญหาของการใช้ browse อยู่ที่การตรวจสอบการกดคีย์ควบคุมระหว่างการป้อนข้อมูล โดยไม่ว่าเคอร์เซอร์จะอยู่ที่ใด ถ้ากดคีย์ควบคุมโปรแกรมจะตรวจสอบคีย์ควบคุมได้และทำงานตาม เช่น การเลื่อนไฮไลต์ขึ้น-ลงไปยังตำแหน่งที่ต้องการ

ดีเบสมีคำสั่ง READKEY( ) สำหรับอ่านคีย์ควบคุม

ในขณะที่ใช้คำสั่ง get เราสามารถเลิกจากคำสั่ง get การเคอร์เซอร์อยู่ที่ใดได้ด้วยการกดคีย์ควบคุม เช่น ↑ ↓ PgUp PgDn ฯลฯ ดีเบสทรีพลัสมีวิธีการตรวจสอบการออกจากเรคอร์ดที่ edit ว่ามีการกดคีย์อะไรมา

รูปที่ 2 การ get ณ ตำแหน่งไฮไลต์ ไฮไลต์ด้วยคำสั่ง get ใช้คำสั่ง say

ชื่อตัวแปร	คีย์ที่กด	ค่าคีย์	ค่าคีย์ที่มีการ update	ความหมาย
rc_bakchr	Ctrl-H Ctrl-S Back space	0		ถอยหลัง 1 ตัวอักษร
rc_update	Ctrl-H Ctrl-S Back space		256	ถอยหลัง 1 ตัวอักษร
rc_forch	Ctrl-D Ctrl-L	1	257	เดินหน้า 1 ตัวอักษร
rc_bakwrd	Ctrl-A	2	258	ถอยหลัง 1 คำ
rc_forwrd	Ctrl-F	3	259	เดินหน้า 1 คำ
rc_bakfld	Ctrl-E Ctrl-K	4	260	ถอยหลัง 1 ฟิลด์
rc_forfld	Ctrl-J Ctrl-X	5	261	เดินหน้า 1 ฟิลด์
rc_baksc r	Ctrl-R	6	262	ถอยหนึ่งจอภาพ
rc_forscr	Ctrl-C	7	263	เดินหน้าหนึ่งจอภาพ
rc_bakpan	Ctrl-Z	8	264	แพนทางซ้าย
rc_forpan	Ctrl-B	9	265	แพนทางขวา
rc_delete	Ctrl-U	10	266	ลบ
rc_insert	Ctrl-N	11	267	แทรก
rc_quit	Ctrl-Q Esc	12	268	เลิกไม่มีการ save
return code		13		รีเทอร์น
rc_write	Ctrl-End Ctrl-W	14	270	เลิกมีการ save
rc_filled	Ctrl-M	15	271	filled, type past ed
rc_return	←┘	16	272	กดรีเทอร์นเมื่อเลิกในฟิลด์แรกเป็น black
rc_menu	Ctrl-Home	33	289	แสดงเมนู on/off
rc_zomout	Ctrl-PgUp	34	290	ซูม out
rc_zomin	Ctrl-PgDn	35	291	ซูม in
rc_help	F1	36	292	help

ตารางที่ 1

แสดงชื่อตัวแปรในดีเบสทรีพลัสที่ใช้เก็บค่าคีย์ควบคุมของคำสั่ง READKEY ( )

คำสั่ง readkey() แตกต่างจาก inkey() เพราะคำสั่ง inkey() อ่านจากบัฟเฟอร์ของคีย์บอร์ด ถ้าไม่มีการกดคีย์จะได้ค่าเป็น 0 แต่คำสั่ง readkey() อ่านจากบัฟเฟอร์ของสกรีนในคำสั่ง get และจะอ่านค่าคีย์สุดท้ายหลังจากที่ได้ผ่านคำสั่ง read มาแล้ว โดยค่าคีย์ที่อ่านได้แม้จะเป็นคีย์เดียวกันก็สามารถอ่านได้สองแบบ คือ มีค่าตัวเลขอยู่ระหว่าง 0 ถึง 36 หรือเกิน 256

ค่าคีย์ที่อ่านได้จากคีย์ควบคุมต่าง ๆ จะให้ค่าแตกต่างจากคำสั่ง inkey() โดยค่าที่ได้นั้น เราอาจนำมาเก็บไว้ในตัวแปรเพื่อตรวจสอบในโปรแกรมได้ โดยดีเบสทรีพลัสกำหนดชื่อตัวแปรให้มีความหมายเป็นที่เข้าใจได้ง่าย เช่นในตารางที่ 1

ในการกดคีย์ฟังก์ชันลูกศรทางขวามือของคีย์บอร์ดนั้น จะมีค่าเท่าเทียมกับการกดคีย์ควบคุมดังนี้

→	Ctrl-D
←	Ctrl-S
↑	Ctrl-E
↓	Ctrl-X
Ctrl- →	Ctrl-B
Ctrl- ←	Ctrl-Z
Ins	Ctrl-V
Del	Ctrl-G
Home	Ctrl-A
End	Ctrl-F
PgUp	Ctrl-R
PgDn	Ctrl-C
Ctrl-Home	Ctrl-]
Ctrl-End	Ctrl-W
Ctrl-PgUp	Ctrl--
Ctrl-PgDn	Ctrl-^

ในการเขียนโปรแกรมสั่งงาน เราสามารถตรวจสอบคีย์ได้ด้วยตัวอย่างต่อไปนี้

```
rc_write = 14
rc_update = 256
If readkey() = rc_write +
   rc_update
   Replace Name With Mname
EndIf
```

ในกรณีนี้หมายถึงให้ตรวจสอบว่ามีการกดคีย์ Ctrl-End หรือไม่ ถ้ามีก็ให้แทนฟิลด์ Name ด้วยตัวแปร Mname ซึ่งเราสามารถใช้

```
If readkey() = 270
```

เป็นตัวเลขแทนเลยก็ได้ สังเกตว่าการใช้ readkey() นี้ จะทำให้เราสามารถใช้ตรวจสอบการกดคีย์ควบคุมในขณะโปรแกรมทำงานแบบฟูลสกรีนได้ นั่นเป็นข้อดีเด่นอย่างหนึ่งของดีเบสทรีพลัส

สร้างโปรแกรมสำหรับแก้ไขข้อมูลแบบใช้งานได้ทั่วไป

มุมดีเบสเป็นเรื่องของเทคนิควิธีการใช้งาน ผู้เขียนจึงพยายามคิดหาวิธีการใช้งานแบบต่าง ๆ มาสร้างเป็นโปรแกรมให้ โดยเน้นรูปแบบโปรแกรมแบบใช้งานได้ทั่วไป ซึ่งเหมาะที่จะนำไปใช้ร่วมกับโปรแกรมอื่น

โปรแกรมนี้เริ่มจากการเขียนจอภาพและ get mfile ซึ่ง mfile คือ ตัวแปรที่จะเก็บชื่อไฟล์ที่นำมา browse นั่นเอง เมื่อผู้ป้อนป้อนชื่อให้แล้ว ก็จะตรวจสอบดูว่ามีไฟล์นี้จริงหรือไม่ ถ้าไม่มีก็จะตอบว่า 'No File' แล้วเลิกงาน

หากตรวจสอบพบว่ามีอยู่ในดิสก์ เราจะเรียกไฟล์นี้ออกมา โดยใช้คำสั่ง use &mfile และก๊อบปี้ไปไว้ยังชื่อ tstru ซึ่งการก๊อบปี้นี้ทำแบบ extended คือ ให้โครงสร้างเป็นข้อมูลของแฟ้มชื่อ tstru หลังจากนั้น จะ use tstru ในพื้นที่ที่ 9

แฟ้ม tstru เป็นแฟ้มที่เก็บโครงสร้างของแฟ้มที่ใช้งาน browse ซึ่งเราสามารถตรวจสอบความกว้าง ชนิด และชื่อฟิลด์ ได้หมด

ในการเริ่มนำข้อมูลมา browse นี้ ถ้าหากว่าข้อมูลไม่มีเลยแม้แต่เรคอร์ดเดียว เราจะใส่เรคอร์ดว่างไว้ให้ 1 เรคอร์ด เสมือนช่วยในขณะที่ต้องการป้อนข้อมูลแบบ browse

การเขียนชื่อฟิลด์และกำหนดความกว้างของฟิลด์ จะทำเองอย่างอัตโนมัติ เพราะโปรแกรมสามารถตรวจสอบได้ว่าจะใช้ความกว้างแต่ละฟิลด์เท่าใด สำหรับโปรแกรมที่เขียนขึ้นนี้ ยังไม่มีการตรวจสอบความยาวของเรคอร์ดเกินจอภาพ ดังนั้น จึงไม่สามารถใช้กับไฟล์ที่ยาวเกินกว่า 80 คอลัมน์ได้ อย่างไรก็ตาม ผู้ใช้งานอาจประยุกต์เพิ่มเติมหรือเลือกเอาบางฟิลด์ก็ได้

โปรแกรม do while ลูปใหญ่สุดเป็นลูปการสแกน ทำ browse โดยมีลูปในหลายลูป เช่น ลูปสำหรับการเขียนข้อมูล ลูปการ get เป็นต้น ส่วนสำคัญของการ get เป็นจุดที่น่าสนใจและเป็นวิธีการที่น่าจำไปใช้ได้ โดย get อยู่ใน do while และ read อยู่นอก do while

โปรแกรมอ่านคีย์ควบคุมขณะ GET ได้อย่างไร

ขณะ get จะเป็นฟูลสกรีนอีดิต ในขณะนี้ดีเบสรับคีย์ควบคุมอยู่แล้ว โดยดีเบสยอมให้ใช้คีย์ควบคุมเหมือนคำสั่ง append แต่เนื่องจากอยู่ภายใต้โปรแกรมของเราเอง คีย์ควบคุมบางคีย์จึงไม่เหมือนสภาพการใช้งานจริง เช่น ถ้ากด ^W หรือ ^END ก็เพียงแต่ทำให้ออกจาก get หรือดีเบสจะ read เลยนั่นเอง ดังนั้นโปรแกรมต้องตรวจสอบเอง

จากโปรแกรมในรูปที่ 3 นี้ เราตรวจสอบคีย์ ^END (^W) คีย์ PgDn, PgUp และคีย์ ↑ ขณะอยู่ที่คอลัมน์แรกหรือไม่ได้แก้ไขข้อมูลเพื่อเลื่อนไฮไลต์ ลักษณะของการเขียนโปรแกรมจะเป็น

รูปที่ 3

```
. TYPE BROWS.PRG
*************************************************
*dBCORNER Microcomputer Journal 34              *
*Program Browse                                 *
*Date : 09-20-87                                *
*Version 0.00001                                *
*All Wrong Reserve ©1987 by KU. Micro. Lab.     *
*************************************************
clear
set talk off
set stat off
set safety off
clear all
clear
sele 1
@ 0,0 say 'Browse.Prg'
@ 0,68 say str(day(date()),2)+' '+left(cmonth(date()),3)+' '+str(year(date()),4)
@  3,  0  TO  3, 78    DOUBLE
@ 4,0 clear
@ 5,30 say 'Browse by dbase programming'
@ 8,1 to 8,78
mfile=space(8)
@ 15,20 say 'Enter file to browse :' get mfile
read
mvar=trim(mfile)+'.dbf'
if .not. file('&mvar')
 @ 9,0 clear
 @ 15,10 say 'No file'
else
 @ 9,0 clear
 use &mfile
 copy to tstru stru extended
 sele 9
 use tstru
 col=1
 do while .not. eof()
  @ 9,col say  field_name
  col=col+field_len+2
  skip
 enddo
 @ 10,1 to 10,78
 @ 23,10 say 'Use arrow key to control ... Ctrl-End to Exit'
 @ 24,9 say 'PgUp for record back .. PgDn for record forward'
 sele 1
 go top
 if reccount()=0
  appe blank
 endif
 rk=1
 mark=0
 do while .not. eof()
  ko=1
  @ 11,0 clear to 22,79
  if mark=1
   mk=rk-9
   goto mk
  else
   goto rk
  endif
  do while .not. eof().and. ko<=10
   col=1
   sele 9
   go top
   do while .not. eof()
    xx=trim(field_name)
    yy=field_len
    sele 1
```

รูปที่ 3 (ต่อ)

```
    @ 10+ko,col say &xx
    col=col+yy+2
    sele 9
    skip
   enddo
   sele 1
   skip
   ko = ko+ 1
  enddo
  col=1
  sele 1
  goto rk
  ko = 1
  do while .not. eof() .and. ko<=10
   goto rk
   sele 9
   col=1
   go top
   do while .not. eof()
    xx=trim(field_name)
    yy=field_len
    sele 1
    if mark=1
     ko=10
     mark=0
    endif
    @ 10+ko,col get &xx
    col=col+yy+2
    sele 9
    skip
   enddo
   sele 1
   read
   sele 9
   col=1
   go top
   do while .not. eof()
    xx=trim(field_name)
    yy=field_len
    sele 1
    @ 10+ko,col say &xx
    col=col+yy+2
    sele 9
    skip
   enddo
   sele 1
   if readkey()=270
    go bottom
    skip
    exit
   endif
   if readkey()=4 .or. readkey()=262 .or. readkey()=6
    rk=rk-1
    ko=ko-1
    if rk=0
     rk=1
     ko=1
    endif
    if ko=0 .and. rk>=10
     ko=1
     mark=1
     exit
    endif
   else
    skip
    if eof()
     ans='N'
```

รูปที่ 3 (ต่อ)

```
      @ 22,10 say 'Do you want to add more record Y/N..' get ans
      read
      if upper(ans)="Y"
       appe blank
      else
       ans='Y'
       @ 22,0 clear to 22,79
       @ 22,0 say 'Do you want to leave this browse..' get ans
       read
       if upper(ans)<>'Y'
        skip -1
        rk=rk-1
        ko=ko-1
        @ 22,0 clear  to 22,79
       endif
      endif
     endif
     rk=rk+1
     ko=ko+1
    endif
   enddo
  enddo
endif
@ 4,0 clear
@ 10,30 say 'B Y E  B Y E .......'
SELE 9
USE
sele 1
use
return
*End program
```

ดังนี้

```
If readkey() = 270 && ^END
  go bottom
  skip && eof active
  exit
endif
```

อีกส่วนหนึ่งคือ การเลื่อนขึ้นด้วย PgUp และ ↑ ขณะไม่ update เราใช้

```
if readkey()   = 4
.or. readkey() = 262
.or. readkey() = 6
```

รหัส 4 คือ ↑ รหัส 262 และ 6 คือ PgUp ส่วน PgDn นั้น ไม่ต้องตรวจสอบเพราะเสมือนเป็นการออกจาก get โปรแกรมก็จะไปทำเรคอร์ดต่อไปอยู่แล้ว

ตัวอย่างของโปรแกรมแสดงในรูปที่ 3

ลองสร้างไฟล์ข้อมูลในที่นี้ขอชื่อ dem โดยมีฟิลด์ต่าง ๆ ดังรูปที่ 4

```
. USE DEM
. LIST STRU
Structure for database: B:DEM.dbf
Number of data records:      14
Date of last update    : 09/20/87
Field  Field Name  Type        Width     Dec
    1  F1          Character      10
    2  F2          Character      10
    3  F3          Numeric        10
    4  F4          Numeric         8       2
** Total **                       39

. LIST
Record#  F1          F2                   F3        F4
      1  test1       testdata1            66     11.00
      2  test2       testdata2            40      2.00
      3  test3       testdata3            33     33.00
      4  test4       testdata4            63     29.00
      5  test5       testdata5            33     55.00
      6  test6       testdata6            44     55.00
      7  test7       testdata7            45     23.00
      8  test8       testdata8            44     44.00
      9  test9       testdata9           333     55.00
     10  test10      testdata10           22     22.00
     11  test11      testdata11            2     33.00
     12  test12      testdata12          222     33.00
     13  test13      testdata13           38     22.00
     14  test14      testdata14            1      1.00
```

รูปที่ 4 ไฟล์ข้อมูล DEM.DBF ที่ทดลองสร้างขึ้น

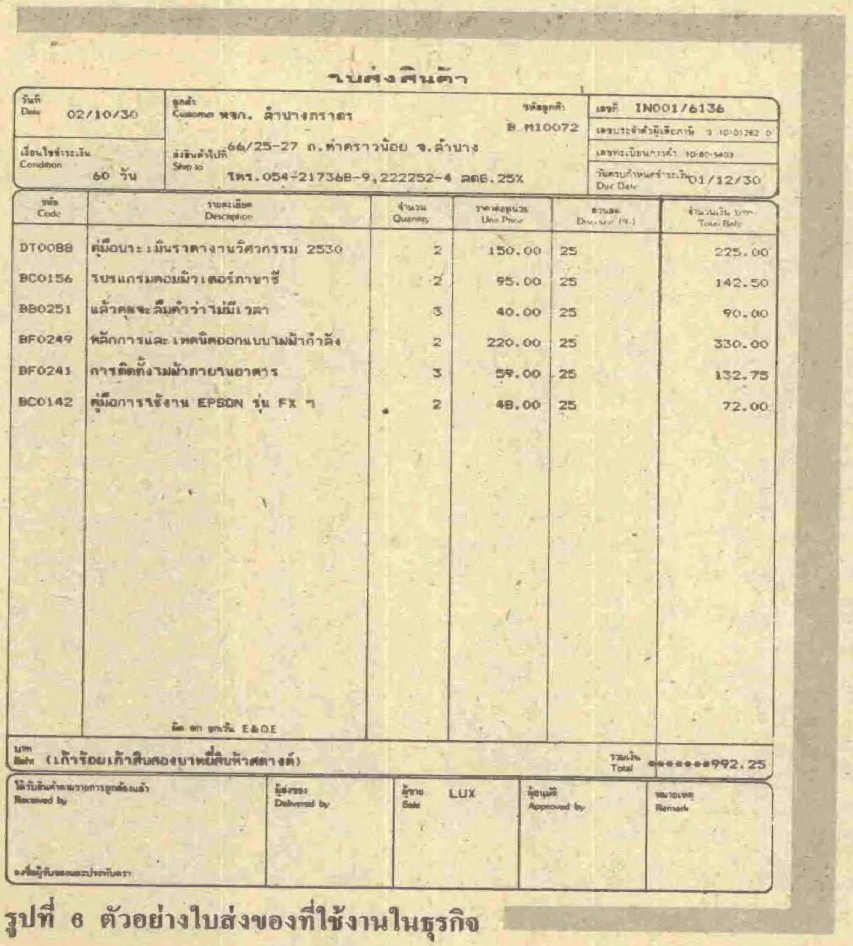
```
Browse.Prg                                   Caps          14 Sep 1987

                      Browse by dbase programming

F1          F2          F3          F4
test1       testdata1       66          11.00
test2       testdata2       40           2.00
test3       testdata3       33          33.00
test4       testdata4       63          29.00
test5       testdata5       33          55.00
test6       testdata6       44          55.00
test7       testdata7       45          23.00
test8       testdata8       44          44.00
test9       testdata9      333          55.00
test10      testdata10      22          22.00

      Use arrow key to control ... Ctrl-End to Exit
    PgUp for record back .. PgDn for record forward
```

รูปที่ 5 ลักษณะของจอภาพเมื่อใช้โปรแกรมในรูปที่ 3 ป้อนและแก้ไขข้อมูลในไฟล์ DEM.DBF ในรูปที่ 4



ใบส่งสินค้า

วันที่ Date 02/10/30 — ลูกค้า Customer หจก. ลำปางการเรา — รหัสลูกค้า B M10072 — เลขที่ IN001/6136

เงื่อนไขชำระเงิน Condition 60 วัน — ส่งสินค้าไปที่ Ship to 66/25-27 ถ.ท่าคราวน้อย จ.ลำปาง โทร.054-217368-9,222252-4 ภพ.25% — วันครบกำหนดชำระเงิน Due Date 01/12/30

รหัส Code	รายละเอียด Description	จำนวน Quantity	ราคาต่อหน่วย Unit Price	ส่วนลด Discount (%)	จำนวนเงิน บาท Total Baht
DT0088	คู่มือประเมินราคางานวิศวกรรม 2530	2	150.00	25	225.00
BC0156	โปรแกรมคอมพิวเตอร์ภาษาซี	2	95.00	25	142.50
BB0251	แล้วคุณจะลืมคำว่าไม่มีเวลา	3	40.00	25	90.00
BF0249	หลักการและเทคนิคออกแบบงานผ้าถักลึง	2	220.00	25	330.00
BF0241	การตัดทั้งงานผ้าภายนอกอาคาร	3	59.00	25	132.75
BC0142	คู่มือการใช้งาน EPSON รุ่น FX ๆ	2	48.00	25	72.00

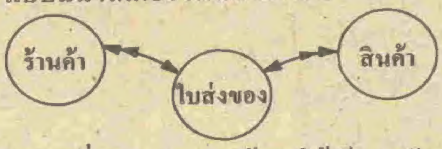
จัดทำโดย ธุรกิจ E&OE

บาท Baht (เก้าร้อยเก้าสิบสองบาทยี่สิบห้าสตางค์) — รวมเงิน Total ๙๙๙

วันที่ และร้านค้าที่ส่ง จากนั้นจึงป้อนข้อมูลเป็นรายการไป

เพื่อให้เห็นตัวอย่างง่ายขึ้น จึงสมมติใบส่งของหนึ่งแบบ โดยมีส่วนหัวประกอบด้วย เล่มที่..... เลขที่..... ลูกค้า..... วันที่ส่ง..... และเป็นตารางต่อมาประกอบด้วย สินค้า จำนวน ราคาต่อหน่วย และจำนวนเลข ในตอนท้ายจะเป็น รวมเงิน

ลักษณะโครงร่างข้อมูลที่ใช้ จะใช้แบบมีมาสเตอร์ไฟล์ก็ได้ เช่น



ซึ่งการบอกลูกค้าจะใช้เพียงรหัสลูกค้า สินค้าก็จะบอกเพียงรหัสสินค้าเท่านั้น ดังนั้น โครงสร้างของแฟ้มใบส่งของนี้จึงมีโครงร่างดังรูปที่ 7

```
. USE DELIVERY INDEX DELIVERY
. LIST STRU .
Structure for database: B:DELIVERY.dbf
Number of data records:        12
Date of last update    : 09/14/87
Field  Field Name  Type        Width    Dec
    1  BOOK_NO     Character       4
    2  DEL_NO      Character       5
    3  ID_CUST     Character       4
    4  DEL_DATE    Date            8
    5  ID_PROD     Character       5
    6  QTY         Numeric         8      2
    7  PRI_UNIT    Numeric        10      2
    8  AMT         Numeric        10      2
** Total **                       55
```

รูปที่ 7 ตัวอย่างโครงสร้างของใบส่งของ

อนึ่ง AMT คือ QTY*PRI_UNIT จะไม่มีก็ได้ แต่ในที่นี้ใส่ไว้ให้ แต่ดีเบสจะคำนวณให้

ลักษณะการป้อนข้อมูล เราจะป้อนแบบ browse ดังตัวอย่างจอภาพในรูปที่ 8 ที่ปรากฏเหมือนใบส่งของ

แฟ้มข้อมูล delivery นี้ เรา index ด้วย book_no + del_no โดยตั้งชื่อเป็น delivery อีก

ในการป้อนนี้ ถ้าหากต้องการแก้ไขเรคอร์ดใดที่อยู่ข้างบน เราก็เลื่อนไฮไลต์ไปแก้ไขได้ด้วยหลักการของ browse หรือถ้าหากจะลบเรคอร์ดใดทิ้ง (ส่วนของสินค้า) ก็เพียงทำให้ช่อง product ว่าง โปรแกรมจะไม่นำไปใส่ในแฟ้ม delivery ให้ ก็เท่ากับเป็นการลบ เพราะโปรแกรมนี้ไม่ตรวจสอบ ^U เพื่อลบเรคอร์ดเหมือน browse จริง ๆ ในดีเบส แต่ถ้าใครอยากจะทำเองก็ได้ ด้วยการเพิ่มการตรวจสอบ ^U แล้วลบเรคอร์ดทิ้ง ตัวเลขทางขวามือและ total เกิด

```
Delivery Order                                    Caps          14 Sep 1987
                          D E L I V E R Y   O R D E R

    Book No. : 1000                  D/O No. : 12350

    Customer ID. : S006              Date of Delivery : 09/14/87

ID_PROD  QTY          PRI_UNIT
P0003      21.00         3.00                  63.00
P0012       2.00        90.00                 180.00
V0001      12.00         8.00                  96.00

                                    TOTAL =     339.00
                                    ==================
```

รูปที่ 8 ลักษณะของจอภาพจะคล้ายใบส่งของ

จากการคำนวณด้วยดีเบส

เมื่อป้อนข้อมูลเสร็จแล้ว ข้อมูลในแฟ้ม delivery จะเป็นดังรูปที่ 9

ตัวโปรแกรมของการจัดการป้อนข้อมูลนี้ได้แสดงไว้ในรูปที่ 10 และเป็นโปรแกรมที่จัดการแฟ้ม delivery เพียงแฟ้มเดียว ซึ่งถ้าหากต้องการเชื่อมโยงกับสินค้าและลูกค้าก็ทำเพิ่มเติมต่อได้ (เพื่อให้ได้ชื่อสินค้า, ลูกค้าเต็ม ๆ)

ส่งท้าย

โปรแกรมทั้งหมดได้เตรียมไว้แล้วใน Softmail ฉบับนี้ ซึ่งผู้เขียนได้จัดทำพร้อมตัวอย่างข้อมูล ผู้สนใจติดต่อมาที่ซีเอ็ดได้ อย่างน้อยก็จะได้ไม่เสียเวลาคีย์และลอกไปใช้กับโปรแกรมอื่น ๆ ได้อีก โดยเฉพาะงานที่ผู้อ่านสร้างขึ้นเอง และใช้ส่วนของโปรแกรม browse ที่ผู้เขียนยกตัวอย่างนี้

พบกันใหม่ด้วยเทคนิคของดีเบสทรีพลัส ในฉบับหน้า

```
. LIST
Record#  BOOK_NO DEL_NO ID_CUST DEL_DATE ID_PROD      QTY    PRI_UNIT      AMT
      1  1000    12345  S001    09/20/87 P0001      2.00        5.00    10.00
      2  1000    12345  S001    09/20/87 P0002     23.00        3.00    69.00
      3  1000    12346  S002    09/20/87 P0003     24.00        9.00   216.00
      4  1000    12346  S002    09/20/87 P0004      2.00        9.00    18.00
      5  1000    12347  S003    09/20/87 P0004      4.00        6.00    24.00
      6  1000    12348  S005    09/20/87 P0002      3.00        5.00    15.00
      7  1000    12349  S006    09/20/87 P0002     12.00        5.00    60.00
      8  1000    12349  S006    09/20/87 P0004     16.50       50.00   825.00
      9  1000    12349  S006    09/20/87 P0005     17.00        7.00   119.00
     10  1000    12350  S006    09/14/87 P0003     21.00        3.00    63.00
     11  1000    12350  S006    09/14/87 P0012      2.00       90.00   180.00
     12  1000    12350  S006    09/14/87 V0001     12.00        8.00    96.00
```

รูปที่ 9 แฟ้ม DELIVERY.DBF เมื่อป้อนข้อมูลเสร็จแล้ว

รูปที่ 10 โปรแกรมช่วยป้อนข้อมูลใบส่งของ ที่มีลักษณะคล้ายใบส่งของจริง

```
**************************************************
*dBASE Corner Microcomputer Journal 34           *
*Microcomputer research lab.                     *
*Program :Enter.prg for Delivery file            *
*Create Date: 08-26-87                           *
*Version :0.001                                  *
*All Wrong Researve @1987                        *
**************************************************
clear all
set talk off
set safety off
clear
sele 3
use delivery index delivery
moreone='Y'
do while upper(moreone)='Y'
 sele 3
 clear
 @ 0,0 say 'Delivery Order '
 @ 0,68 say str(day(date()),2)+' '+left(cmonth(date()),3)+' '+str(year(date()),4)
 @  3,  0  TO  3, 78
 mbook_no = space(4)
 mdel_no=space(5)
 @ 2,30 say 'D E L I V E R Y  O R D E R '
 @ 5,5 say 'Book No. :' get mbook_no
 @ 5,35 say 'D/O No. :' get mdel_no
 @ 24,10 say 'Enter blank or return key ... to exit..'
 read
 @ 24,0 clear
```

รูปที่ 10 (ต่อ)

```
mkey=mbook_no+mdel_no
if mkey=space(1)
 @ 4,0 clear
 @ 10,10 say 'No enter...'
 @ 20,0
 wait
 moreone='N'
else
 seek mkey
 if found()
  @ 4,0 clear
  @ 10,10 say 'This data already exist.....'
  @ 20,0
  wait
 else
  mid_cust=space(4)
  mdate=date()
  @ 7,5 say 'Customer ID. :' get mid_cust
  @ 7,35 say 'Date of Delivery :' get mdate
  read
  sele 8
  use
  sele 3
  copy to temp stru field id_prod,qty,pri_unit
  sele 9
  use temp
  copy to tstru stru extended
  use tstru
  sele 8
  use temp
  sele 9
  col=1
  do while .not. eof()
   @ 9,col say  field_name
   col=col+field_len+4
   skip
  enddo
  @ 10,1 to 10,78
  sele 8
  go top
  if reccount()=0
   appe blank
  endif
  rk=1
  mark=0
  do while .not. eof()
   ko=1
   @ 11,0 clear to 21,78
   @ 10,0 to 20,0
   @ 10,7 to 20,7
   @ 10,20 to 20,20
   @ 10,34 to 20,34
   if mark=1
    mk=rk-9
    goto mk
   else
    goto rk
   endif
   do while .not. eof().and. ko<=10
    col=1
    sele 9
    go top
    do while .not. eof()
     xx=trim(field_name)
     yy=field_len
     sele 8
```

รูปที่ 10 (ต่อ)

```
   @ 10+ko,col say &xx
   col=col+yy+4
   sele 9
   skip
  enddo
  sele 8
  skip
  ko = ko+ 1
enddo
col=1
sele 8
goto rk
ko = 1
do while .not. eof() .and. ko<=10
 goto rk
 sele 9
 col=1
 go top
 do while .not. eof()
  xx=trim(field_name)
  yy=field_len
  sele 8
  if mark=1
   ko=10
   mark=0
  endif
  @ 10+ko,col get &xx
  col=col+yy+4
  sele 9
  skip
 enddo
 sele 8
 read
 NP=RECNO()
 amt=qty*pri_unit
 @ 10+ko,50 say str(amt,10,2)
 go top
 tot =0
 do while .not. eof()
  tot=tot+qty*pri_unit
  skip
 enddo
 @ 23,45 say 'TOTAL =' +str(tot,10,2)
 @ 24,45 say '==================='
 goto np
 sele 9
 col=1
 go top
 do while .not. eof()
  xx=trim(field_name)
  yy=field_len
  sele 8
  @ 10+ko,col say &xx
  col=col+yy+4
  sele 9
  skip
 enddo
 sele 8
 if readkey()=270
  go bottom
  skip
  exit
 endif
 if readkey()=4 .or. readkey()=262 .or. readkey()=6
  rk=rk-1
  ko=ko-1
```

รูปที่ 10 (ต่อ)

```
      if rk=0
       rk=1
       ko=1
      endif
      if ko=0 .and. rk>=10
       ko=1
       mark=1
       exit
      endif
     else
      skip
      if eof()
       ans='Y'
       @ 22,1 say 'Do you want to add more lineitem Y/N..' get ans
       read
       if upper(ans)="Y"
        appe blank
       else
        ans='Y'
        @ 22,0 clear to  22,75
        @ 22,0 say 'Do you want to leave this delivery order..' get ans
        read
        if upper(ans)<>'Y'
         skip -1
         rk=rk-1
         ko=ko-1
         @ 22,0 clear  to 22,75
        endif
       endif
      endif
      rk=rk+1
      ko=ko+1
     endif
    enddo
   enddo
   @ 4,0 clear
   @ 10,30 say 'W A I T . . . .'
   SELE 8
   go top
   do while .not. eof()
    sele 3
    if temp->id_prod=space(1)
     sele 8
     skip
    else
     appe blank
     repl book_no with mbook_no
     repl del_no with mdel_no
     repl id_cust with mid_cust
     repl del_date with mdate
     repl id_prod with temp->id_prod
     repl qty with temp->qty
     repl pri_unit with temp->pri_unit
     repl amt with qty*pri_unit
     sele  8
     skip
    endif
   enddo
  endif
 endif
enddo
sele 8
use
sele 9
use
return
```